Princeton University

MAE-1826

ARTIFICIAL INTELLIGENCE THEORY AND RECONFIGURABLE CONTROL SYSTEMS

FINAL REPORT
for
ARMY RESEARCH OFFICE CONTRACT
No. DAAG29-84-K-0048

June 1988

Prepared for:

U.S. ARMY RESEARCH OFFICE
Research Triangle Park, NC 27709

Department of
Mechanical and
Aerospace Engineering

UNCLASSIFIED MASTER COPY - FOR REPRODUCTION PURPOSES

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY 2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
4. PERFORMING ORGANIZATION REPORT NUMBER(S) MAE-1826	5. MONITORING ORGANIZATION REPORT NUMBER(S) ARO 20155.15-MA
6a. NAME OF PERFORMING ORGANIZATION PRINCETON UNIVERSITY 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION U. S. Army Research Office
6c. ADDRESS (City, State, and ZIP Code) Dept. of Mechanical & Aerospace Engineering D202, Engineering Quadrangle Princeton, New Jersey 08544	7b. ADDRESS (City, State, and ZIP Code) P. O. Box 12211 Research Triangle Park, NC 27709-2211
8a. NAME OF FUNDING / SPONSORING ORGANIZATION U. S. Army Research Office 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER DAAG29-84-K-0048
8c. ADDRESS (City, State, and ZIP Code) P. O. Box 12211 Research Triangle Park, NC 27709-2211	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. | PROJECT NO. | TASK NO. | WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
ARTIFICIAL INTELLIGENCE THEORY AND RECONFIGURABLE CONTROL SYSTEMS

12. PERSONAL AUTHOR(S)
Robert F Stengel

13a. TYPE OF REPORT FINAL REPORT	13b. TIME COVERED FROM 3-12-84 TO 3-11-88	14. DATE OF REPORT (Year, Month, Day) June 1988	15. PAGE COUNT 24

16. SUPPLEMENTARY NOTATION
The view, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

17. COSATI CODES		
FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Artificial Intelligence; Control Theory; Flight Control Systems; Helicopter Control Systems; Computer Software; Cybernetics

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This is the final report for a research project entitled "Artificial Intelligence Theory and Reconfigurable Control Systems" (ARO Contract No. DAAG29-84-K-0048). The principal topic of research in this effort was the development of analysis and design methods for fault-tolerant control systems, using artificial intelligence concepts to screen failure hypotheses, and more conventional control theory for reconfiguration and actual control. Methods developed include new graphical presentations of logic trees and signal dependency graphs, procedures for translating expert systems from LISP to Pascal, the definition of performance metrics for

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT □ UNCLASSIFIED/UNLIMITED □ SAME AS RPT. □ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code)	22c. OFFICE SYMBOL

83 APR edition may be used until exhausted.
All other editions are obsolete.

real-time expert systems, a Control Equation Parser for computer-aided control system design, and a control-reconfiguration strategy based on proportional-integral implicit model-following. A Rule-Based Flight Control System for a tandem-rotor helicopter, implemented with parallel microprocessors and operating in real time, was developed in the course of this research.

A sub-task entitled "Computer-Aided Heuristics for Piloted Flight" also was accomplished with funds made available by the Naval Air Systems Command of the U.S. Navy. The purpose of this task was to identify an expert system structure that would reduce the pilot workload associated with flying a single-seat high-performance aircraft. This research culminated in the definition of AUTOCREW, an "expert system of expert systems" based on the paradigm of a World War II bomber crew. AUTOCREW was implemented in a non-real-time personal computer simulation, and its structure is well-posed for parallel implementation.

Both tasks made use of the Princeton Rule-Based Controller, a unique software architecture for combining procedural and symbolic processing. The knowledge base is developed in the LISP computer language and is translated with the inference engine into the Pascal language. The ease with which Pascal code can be embedded in the knowledge-base structure makes this a highly suitable tool for task and algorithm scheduling.

MAE-1826

ARTIFICIAL INTELLIGENCE THEORY AND RECONFIGURABLE CONTROL SYSTEMS

FINAL REPORT
for
ARMY RESEARCH OFFICE CONTRACT
No. DAAG29-84-K-0048

June 1988

Prepared for:

U.S. ARMY RESEARCH OFFICE
Research Triangle Park, NC 27709

Robert F. Stengel
Professor
Principal Investigator

PRINCETON UNIVERSITY
Department of Mechanical and Aerospace Engineering
Princeton, NJ 08544

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED

THE VIEW, OPINIONS, AND/OR FINDINGS CONTAINED IN THIS REPORT ARE THOSE OF THE AUTHOR(S) AND SHOULD NOT BE CONSTRUED AS AN OFFICIAL DEPARTMENT OF THE ARMY POSITION, POLICY, OR DECISION, UNLESS SO DESIGNATED BY OTHER DOCUMENTATION.

ABSTRACT

This is the final report for a research project entitled "Artificial Intelligence Theory and Reconfigurable Control Systems" (ARO Contract No. DAAG29-84-K-0048). The principal topic of research in this effort was the development of analysis and design methods for fault-tolerant control systems, using artificial intelligence concepts to screen failure hypotheses, and more conventional control theory for reconfiguration and actual control. Methods developed include new graphical presentations of logic trees and signal dependency graphs, procedures for translating expert systems from LISP to Pascal, the definition of performance metrics for real-time expert systems, a Control Equation Parser for computer-aided control system design, and a control-reconfiguration strategy based on proportional-integral implicit model-following. A Rule-Based Flight Control System for a tandem-rotor helicopter, implemented with parallel microprocessors and operating in real time, was developed in the course of this research.

A sub-task entitled "Computer-Aided Heuristics for Piloted Flight" also was accomplished with funds made available by the Naval Air Systems Command of the U.S. Navy. The purpose of this task was to identify an expert system structure that would reduce the pilot workload associated with flying a single-seat high-performance aircraft. This research culminated in the definition of AUTOCREW, an "expert system of expert systems" based on the paradigm of a World War II bomber crew. AUTOCREW was implemented in a non-real-time personal computer simulation, and its structure is well-posed for parallel implementation.

~~Both tasks~~ made use of the Princeton Rule-Based Controller, a unique software architecture for combining procedural and symbolic processing. The knowledge base is developed in the LISP computer language and is translated with the inference engine into the Pascal language. The ease with which Pascal code can be embedded in the knowledge-base structure makes this a highly suitable tool for task and algorithm scheduling.

The period of performance for this contract was 3/12/84 to 3/11/88.

TABLE OF CONTENTS

1. INTRODUCTION

As computers become smaller, faster, and more reliable, they change the way in which many things are done. Hardware is giving way to software: the functions of machines formerly performed by mechanisms and structures are being taken over by computers and numerical logic. Increased reliance is being placed on electronic systems for control, and it is becoming crucial that these systems operate with a high degree of reliability. Performance must be assured, even when system components fail; otherwise, there is the danger not only of aborting the original mission but of losing the entire machine. The new control systems must not only be reliable by conventional standards, they must be tolerant of a range of failures not previously considered detrimental.

1.1 FAULT-TOLERANT CONTROL SYSTEMS

Fault-tolerant control systems typically have one or more of three attributes: robustness, reconfigurability, and restructurability. *Robustness* implies insensitivity to variations in the parameters or structure of the system. If physical parameters change or the nature of system dynamics is altered, the robust system continues to operate as originall} intended, providing response to inputs and disturbances that is close to nominal performance. *Reconfigurability* implies that some of the control system's parameters can be purposely modified to account for uncontrollable changes in the system, such as failed sensors or actuators or damaged structural clements. *Restructurability* subsumes reconfigurability, implying that not only parameters but the structure of the system itself can be changed to accommodate uncontrollable changes. Whereas the robust system obtains fault tolerance with fixed parameters and structure, reconfigurable and restructurable systems must be adaptive, providing some degree of fault detection and identification as well as the ability to alter the system. The latter should achieve satisfactory performance over a wider range of conditions than is obtained by robustness alone; otherwise the addcd complexity is not warranted.

While the robust system might be considered "dumb," in the sense that it does not have to know about or respond to system changes, restructurable and reconfigurable systems must be "smart," sensing and adjusting to system changes. Consequently, these latter systems must possess a degree of *intelligence* -- an ability to calculate, reason, perceive relationships and analogies, learn, store information, classify, generalize, and adjust to new situations. Of course, they should be robust to the extent that the situation does not get out of hand while adaptation is occurring. Together, intelligence and information form *knowledge*. Systems that restructure or reconfigure can be called *Knowledge-Based Systems*, for they must contain the intelligence to act and the information on which to base needed actions.

1.2 BACKGROUND IN MACHINE INTELLIGENCE AND CONTROL THEORY

Since the term "artificial intelligence" was first coined in 1956, the field has been the home of research in expert systems, automatic programming, theorem proving, perception problems, data-base retrieval, robotics, combinatorial and scheduling problems, and natural language processing [1-3]. The first six of these are most closely related to the proposed research. *Expert systems* use the paradigm of a panel of experts to organize rule-based decision making. Although most are basically deductive in operation, some use inductive reasoning for knowledge acquisition [4]. *Automatic programming* establishes a basis for translating and compiling instructions and data that are specified in one form into another form that is suitable for computation. *Theorem proving* deals with the use of given evidence to obtain a general statement of (usually mathematical) fact. *Perception problems* could address a wide range of issues; in the current context, we are most interested in the effect that *belief systems* may have on the efficiency and correctness of real-time decision making [5]. *Data-base retrieval* is endemic to the use of any knowledge-based system; the principal area of interest here is in efficient search routines that relate failure symptoms to failed elements. *Robotics* treats the application of decision making and control to physical systems; for our purposes, the helicopter may be considered to be the robot.

The relationship between machine intelligence and control theory certainly predates the modern interest in robotics, as Norbert Weiner and others were addressing cybernetics and automata many decades ago [6,7]. Nevertheless, there seemed little reason to connect the two until computers made the necessary computations and communication practical. In fact, it is clear that the present state of robotics owes far more to control theory than to machine intelligence [8].

The development of control theory tended to follow available applications; high-performance aircraft of the 1950s needed stability augmentation, and the technology was available. As aircraft encountered a growing range of flight conditions, adaptation (by scheduling control gains and limits with dynamic pressure and Mach number) became common, introducing a modicum of intelligence to these early systems. A significant portion of the control theory that ultimately will contribute to robotics and intelligent control had its origins in aeronautical applications.

"Fuzzy sets" provide another example of the contribution that control theory has made to machine intelligence [9]. Fuzzy sets have their origins in classical probability and statistics as applied to signal processing and control systems; they assign subjective attributes to uncertain factors and are claimed to provide an "expressive" framework for making decisions with soft information. The fuzzy set approach has been adopted as an alternative to Bayesian logic in some expert systems that reason under uncertainty.

Fault-tolerant control developments have tended to follow the concepts of robustness [10-11], parallel redundancy [12-13], analytical redundancy [14-15], and self-tuning or adaptive regulation [16-17]. Robustness has been discussed previously. *Parallel redundancy* typically uses comparison, voting, and/or averaging of similar components either to select out failed components or to reduce their impact [18]. *Analytical redundancy* combines the remaining unfailed dissimilar components to perform the functions of a failed component. A *self-tuning regulator* uses simple gain-adjustment laws to maintain some system metric (e.g., bandwidth or natural frequency) at a desirable value. The latter three approaches can be said to restructure or reconfigure the controller.

2. RESULTS OF THE PROJECT

2.1 ARTIFICIAL INTELLIGENCE THEORY AND RECONFIGURABLE CONTROL SYSTEMS

The project *Artificial Intelligence Theory and Reconfigurable Control Systems* was begun in March 1984 and was completed in March 1988. The principal goals were to demonstrate a real-time expert system for fault-tolerant control and to investigate methods for fault-tolerant control systems with limited on-line learning attributes. These goals have been met, and additional accomplishments have been achieved. Research results have been documented in five interim technical reports [19-23], one technical note [24], and eight technical papers [25-32]. One M.S.E. thesis

[33] and two Ph.D. theses [34,35] will be completed shortly.

Several interim reports expanded on concepts that were explored in the project. Reference 19 described the general philosophy of the work, presenting the program of research, defining expert and production systems, reviewing control principles, and discussing fault detection and identification. Methods of analytical redundancy were reviewed and their potential combination with rule-based search was described in Ref. 20. A LISP-based expert system for detecting failures in aircraft systems was presented in Ref. 21. References 22 and 23 report the results of an undergraduate project, in which a portion of the maintenance diagnostic procedures described in the Army maintenance manuals for the CH-47 helicopter were transcribed to a LISP-based expert system. Reference 24 summarized the project for a national artificial intelligence newsletter.

The real-time Rule-Based Fault-Tolerant Flight Control System (RBFCS) shown in Fig. 1 has five principal components: Executive Control, Failure Detection, Failure Diagnosis, Failure Model Estimation, and Reconfiguration. Each of these components can be realized as a separate process; for the present multiprocessor implementation, they are grouped in three processes according to computational complexity, as shown in Fig. 2. Each process is resident on a separate 80286/7-based single board computer, and the three processors communicate via a Multibus connection (Fig. 3). The three-board unit plus required analog/digital and digital/analog converters would easily fit inside a single Flight Computer Unit housing used in earlier flight tests at Princeton.

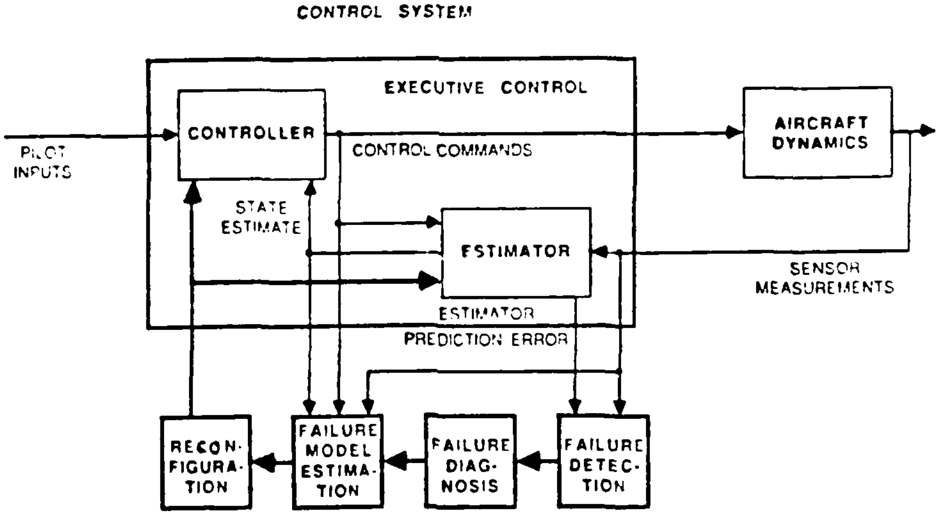


Figure 1. Organization of the Rule-Based Fault-Tolerant Flight Control System.

Our initial approach to combining quantitative and qualitative information for Failure Diagnosis is described in Ref. 25. The dynamic relationships of the controlled system (Fig. 4) are translated into a *Signal Dependency Graph* (Fig. 5). The goal of failure diagnosis is to determine the failure origin from system measurements. Because a limited number of measurements leave some uncertainty regarding the failure origin, the initial objective is to deduce a set of most likely failure-origin hypotheses through a process of *Elimination, Intersection, and Ordering* of failure-origin candidates. The problem reduction follows the search indicated by Fig. 6 using cause-and-effect sensitivities based on worst-case effects of all catalogued failures. Failure Diagnosis serves to screen the large number of possible failures in preparation for Failure Model Determination, which uses a multiple-model algorithm to pin down the most likely failure state.

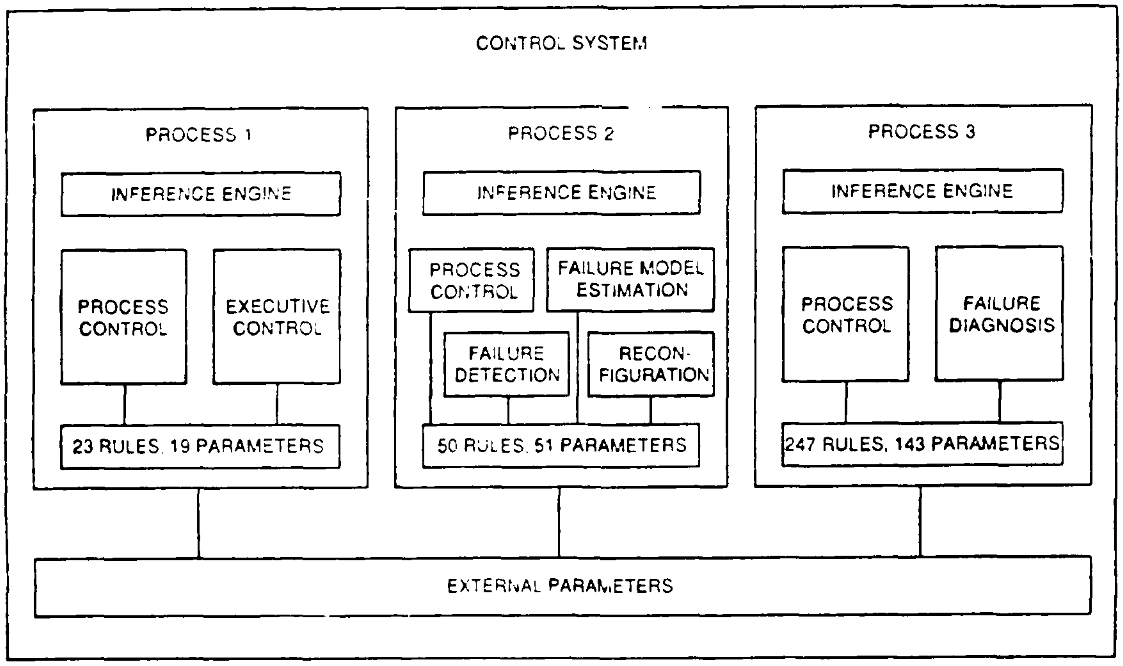


Figure 2. Software Architecture of the Rule-Based Fault-Tolerant Flight Control System.

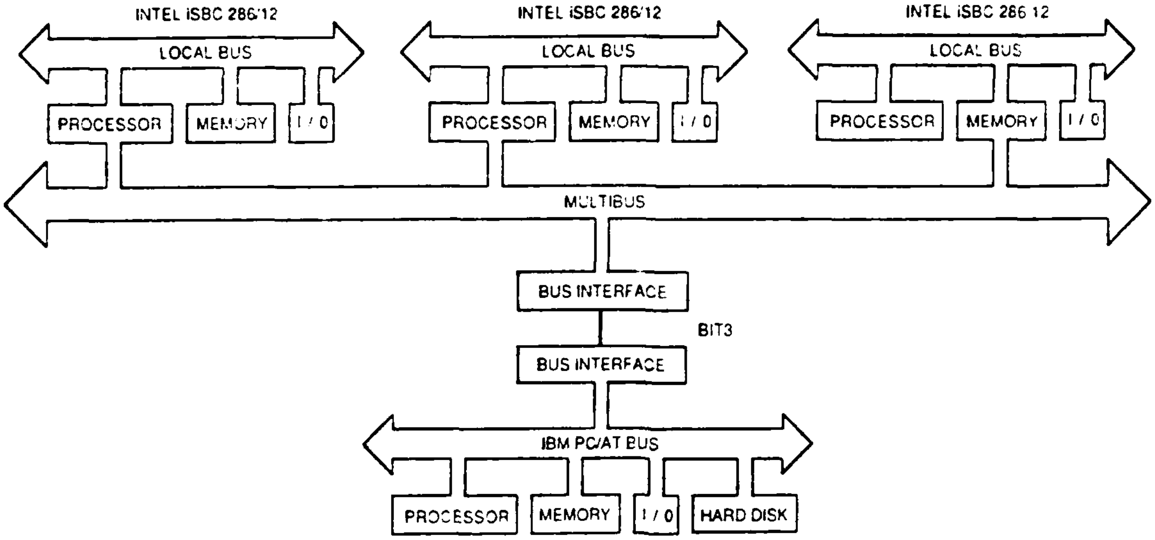


Figure 3. Hardware Architecture of the Rule-Based Fault-Tolerant Flight Control System.

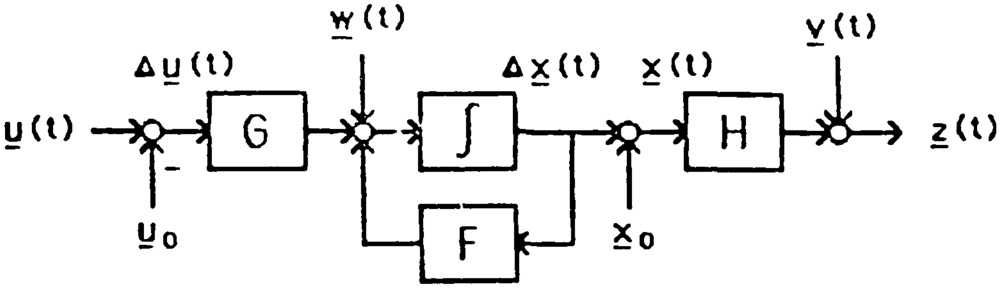


Figure 4. Block Diagram of a Typical Dynamic System.

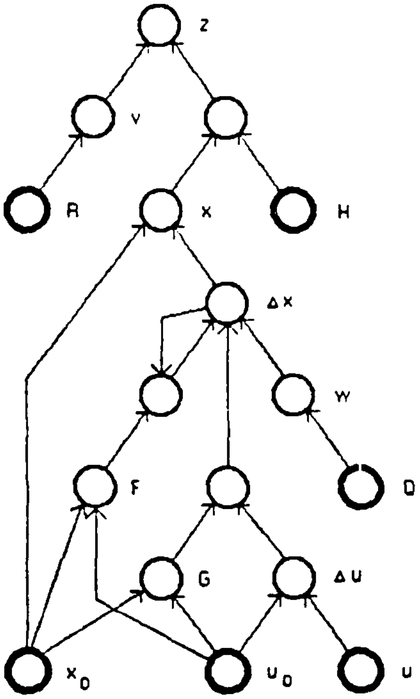


Figure 5. Signal Dependency Graph for a Typical Dynamic System.

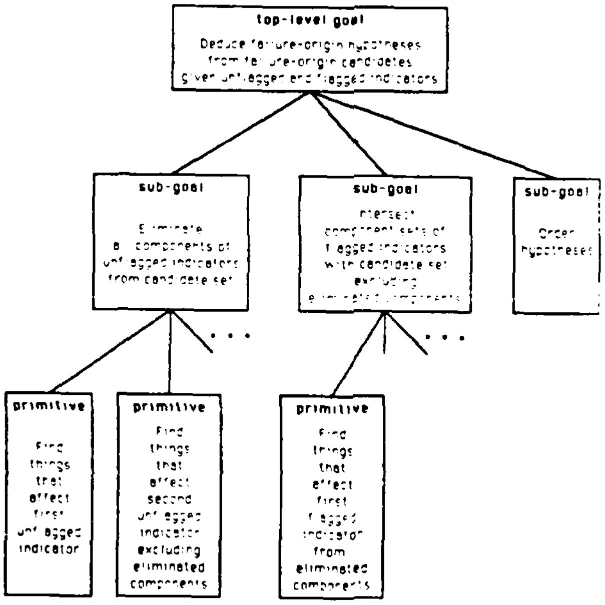

top-level goal
Deduce failure-origin hypotheses from failure-origin candidates given unflagged and flagged indicators
sub-goal
Eliminate all components of unflagged indicators from candidate set
sub-goal
Intersect component sets of flagged indicators with candidate set excluding eliminated components
sub-goal
Order hypotheses
primitive
Find things that effect first unflagged indicator
primitive
Find things that effect second unflagged indicator excluding eliminated components
primitive
Find things that effect first flagged indicator from eliminated components
. . .
. . .

Figure 6. Problem Reduction Approach to Failure-Origin Diagnosis.

Reference 27 presents an overview of the RBFCS, whose contents are summarized in Table 1. We have modified the typical logic diagram that portrays the "firing" of rules and the setting of parameters to explicitly show the parameter values that trigger specific rule actions (Fig. 7). This new symbology provides a much clearer indication of rule-firing effects than do previous notations; an application to the Executive Control Knowledge Base is shown in Fig. 8. The principal functions of the control system, including estimation and control computation, are executed as "side effects" of the search procedure. When a rule is tested, it may initiate a numerical task to obtain the quantitative answers that allow a response of "True." Further developments of the RBFCS were reported in Ref. 29.

Table 1
Control System Knowledge Base Contents

					Time Required		Memory Required Pascal Representation		
Process	Process Tasks	Major Process Numerical Procedures	Number of Process Parameters	Number of Process Rules	Parameter-Rule Association (min)	Knowledge Base Translation (min)	Parameter Data (Kbytes)	Rule Data (Kbytes)	Rule Code (Kbytes)
1	Process 1 Control Executive Control	Kalman Filter Linear Quadratic Regulator	20	24	2.3	3.3	2.4	6.1	2.6
2	Process 2 Control Failure Detection Failure Mode: Estimation Reconfiguration	Normalized Innovations Monitor Multiple-Model Algorithm Weighted Left Pseudo-Inverse	52	51	4.7	5.8	6.2	13.	4.7
3	Process 3 Control Failure Diagnosis		144	248	58.	25.	17.	63.	24.

Pascal inference engine requires 13 Kbytes (excluding compiler library routines)

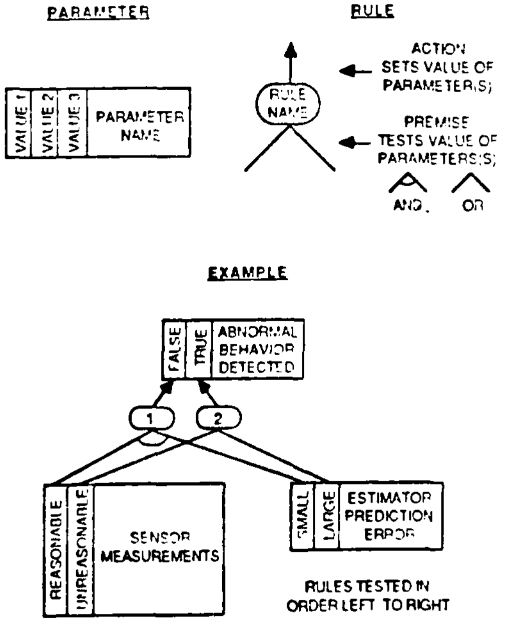

PARAMETER
VALUE 1
VALUE 2
VALUE 3
PARAMETER NAME
RULE
RULE NAME
ACTION SETS VALUE OF PARAMETER(S)
PREMISE TESTS VALUE OF PARAMETERS(S)
AND
OR
EXAMPLE
FALSE
TRUE
ABNORMAL BEHAVIOR DETECTED
1
2
REASONABLE
UNREASONABLE
SENSOR MEASUREMENTS
SMALL
LARGE
ESTIMATOR PREDICTION ERROR
RULES TESTED IN ORDER LEFT TO RIGHT

Figure 7. Graphical Representation of the Knowledge Base.

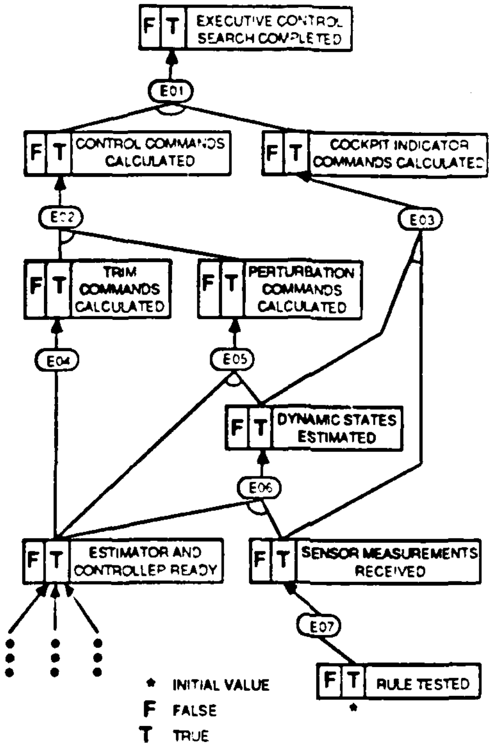

F T EXECUTIVE CONTROL SEARCH COMPLETED
E01
F T CONTROL COMMANDS CALCULATED
F T COCKPIT INDICATOR COMMANDS CALCULATED
E02
E03
F T TRIM COMMANDS CALCULATED
F T PERTURBATION COMMANDS CALCULATED
E04
E05
F T DYNAMIC STATES ESTIMATED
E06
F T ESTIMATOR AND CONTROLLER READY
F T SENSOR MEASUREMENTS RECEIVED
E07
F T RULE TESTED
* INITIAL VALUE
F FALSE
T TRUE

Figure 8. Sample of Executive Control Knowledge Base.

The Control Equation Parser is a computer program for symbolic manipulation of control system design equations [26]. It translates expressions written in standard state-space format into the LISP computer language for evaluation. Vectors and matrices are defined easily, and common operations of linear algebra are executed readily. Numerical solutions then can be obtained by executing LISP code or by calling FORTRAN subroutines. The program provides a highly interactive tool for assisting in the design of multivariable control systems.

Inference mechanisms that deduce the most likely failure mechanisms for given symptoms were presented in Ref. 28. The analysis is carried out in a local sense, where only probabilistic information and causality are used to generate failure models, then in a global sense, where the models are grouped for heuristic pruning. This work is supported by a simulation that models the flight control systems and subsystems of the CH-47 helicopter in substantial detail.

Having determined a failure model, it is necessary to restructure the control logic. If this is to be done at a high level of knowledge, i.e., with-out storing pre-calculated gains for all possible failure states, some degree of on-line control design capability is required. It is especially desirable to maintain command response of the system as close to nominal as possible. One promising approach to achieving this goal is to combine integral compensation (for proper steady-state performance) with implicit model-following (for minimally variant transient response, eigenvalues, and eigenvectors). Linear-quadratic realizations of such systems are explored in Ref. 30.

One important "side effect" of the first program has been the development of an interactive LISP-Pascal translator that runs on the IBM PC-AT computer. With the translator, it is possible to develop an expert system on the PC using LISP; when a satisfactory rule structure has been obtained, new rules are automatically coded in Pascal for fast execution. Both LISP and Pascal expert systems use the same data base, facilitating program development. The compiled Pascal code typically runs over forty times faster than the interpreted LISP code [29].

The three theses to be completed will document the remaining principal results of this project. Several characteristics of model-following controllers and extensions to the Control Equation Parser are presented in the M.S.E. thesis of Parvadha Suntharalingam [33]. The Rule-Based Fault-Tolerant Flight Control System is presented in the Ph.D. thesis of David Handelman [34]. An alternate approach to intelligent restructurable control is presented in the Ph.D. thesis of Chien Huang [35].

2.2 COMPUTER-AIDED HEURISTICS FOR PILOTED FLIGHT

The project *Computer-Aided Heuristics for Piloted Flight* was begun in March 1985 and was completed in March 1988. This was an investigation of the potential for improving the effectiveness and survivability of hazardous flight missions through the use of on-board computation. The principal approach of the study was the development of a symbolic representation for the pilot's functions, incorporation of this model in a computational structure, and simulation of the process. Research results have been documented in five interim technical reports [36-40] and one technical paper [41]. One M.S.E. thesis [42] will be completed shortly.

2.2.1 Cockpit Technology and Simulation

In Ref. 36, Parvadha Suntharalingam explored the features of a voice recognition unit for possible use in the cockpit. The unit was speaker-dependent, that is, it required training with a given vocabulary as spoken by a specific individual. Tests were run with a 50-word vocabulary representative of normal piloting functions to determine the unit's error rates as functions of machine training level and to determine the relative merits of structured and unstructured grammars. It was found that error rates varied from one speaker to the next; some speakers required more training passes than others to achieve a given level of recognition accuracy. It also was noted that a structured grammar required less training than an unstructured one to achieve a given accuracy. However, the latter may be preferred for its greater flexibility; with sufficient training, the error rates may be reduced to acceptable levels.

Brenda Belkin developed a demonstration expert system to evaluate a knowledge acquisition tool for simulating an electronic aid to assist the pilot with in-flight emergency procedures [37]. Four modes of pilot assistance were identified, two requiring manual intervention and two operating autonomously.

John McCaffrey developed a graphics-display simulation of a multi-function display for a high-performance aircraft [38,39]. The display represents a horizontal situation indicator with both "North-Up" and "Aircraft-Heading-Up" modes for use in the cockpit simulator described in Chapter 3.

2.2.2 Cooperative Rule-Based Systems for Aircraft Navigation and Control

Nine cooperating rule-based systems for the combat-aircraft environment were developed in the AUTOCREW expert system [40-42]. Each component system is modeled on a typical World War II bomber crew member having specific task responsibilities. Tasks performed by crew members are easily identified, well-defined, and familiar to operating personnel. The tasks to be performed fall into analogous logical groups.

The AUTOCREW members are responsible for performing tasks and controlling functions associated with the aircraft and its on-board systems. The modeled crew members are COPILOT (flight control), ENGINEER (system monitoring, fault diagnosis, and reconfiguration), NAVIGATOR (navigation), COMMUNICATOR (radio/data operations), OBSERVER (lookout and alarm), ATTACKER (offensive weapon control), DEFENDER (defensive weapon control), and SPOOFER (countermeasures). The ninth rule-based system (EXECUTIVE) functions on a higher level than the others, coordinating mission-specific tasks and addressing overall mission objectives. The human pilot is, of course, in overall command; AUTOCREW works in parallel to his goal-oriented direction.

AUTOCREW knowledge bases are developed and implemented independently. Shared parameters are identified through the design process. These "global" variables are used to exchange information between data bases. A change in value of a shared parameter in one knowledge base can invoke search activity within another knowledge base. The block diagram in Fig. 9 depicts the integration of AUTOCREW within a pilot-vehicle framework. Communication between the pilot and the system would follow typical cockpit-design practice, using graphical and audio displays, manual control devices, and voice input.

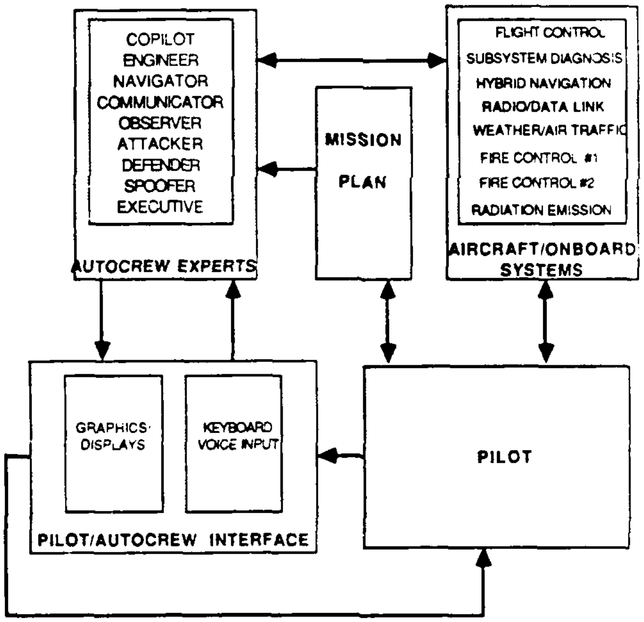


Figure 9. AUTOCREW Configuration.

A summary of the specific tasks associated with AUTOCREW is shown in Table 2. Since this research deals with the design of the system on a macroscopic level, it is sufficient initially to incorporate a general structure into each knowledge base, outlining only major tasks. A graphical representation of a typical crew member's knowledge base (DEFENDER) is shown in Fig. 10. The symbolism follows that presented earlier, with the addition of indicators for shared parameters, dynamic parameters, and "asked for" parameters. Each rule contains a premise and an action; in order for the action statements to be executed, the conditional statements of the premise must be true.

Table 2
Summary of AUTOCREW Tasks and
Skeletal Knowledge-Base Contents

AUTOCREW COMPONENT	MAJOR TASKS	PARAMETERS	RULES
COPILOT	FLIGHT CONTROL, AIRCRAFT PERFORMANCE, TERRAIN FOLLOWING, TERRAIN AVOIDANCE	10	10
ENGINEER	AIRCRAFT SYSTEM DIAGNOSIS, RECONFIGURATION, EMERGENCY PROCEDURES	6	6
NAVIGATOR	NAV SENSOR MANAGEMENT, NAV ERROR STATE ESTIMATION, DYNAMIC ROUTE PLANNING	18	23
COMMUNICATOR	RADIO OPERATIONS, DATA LINK	5	5
OBSERVER	LOOKOUT, ALARM: ● weather, air traffic ● inbound armament IDENTIFICATION FRIEND OR FOE	13	12
ATTACKER	OFFENSIVE WEAPONS STORES MANAGEMENT, OFFENSIVE FIRE CONTROL SYSTEM, TARGET ACQUISITION/PRIORITIZATION	13	12
DEFENDER	DEFENSIVE WEAPONS STORES MANAGEMENT, DEFENSIVE FIRE CONTROL SYSTEM	13	9
SPOOFER	ELECTRONIC MEASURES, ELECTRONIC COUNTERMEASURES	5	5
EXECUTIVE	MISSION PLANNING, COMMAND, SPECIALIZED TASK COORDINATION	12	13

The Princeton Rule-Based Controller (PRBC) used here and above is a unique software architecture for combining procedural and symbolic processing. The knowledge base is developed in the LISP computer language and is translated with the inference engine into the Pascal language. The ease with which Pascal code can be embedded in the knowledge-base structure makes the PRBC a highly suitable tool for task and algorithm scheduling. The PRBC inference engine uses a goal-directed, depth-first search to operate on a knowledge base. In depth-first search, the left-most branch of an AND/OR graph is first searched exhaustively. If the value of a goal parameter is required, the inference engine proceeds down a branch until enough information is inferred to determine the value of the goal parameter. The graphical representation of each knowledge base facilitates tracing the logic flow.

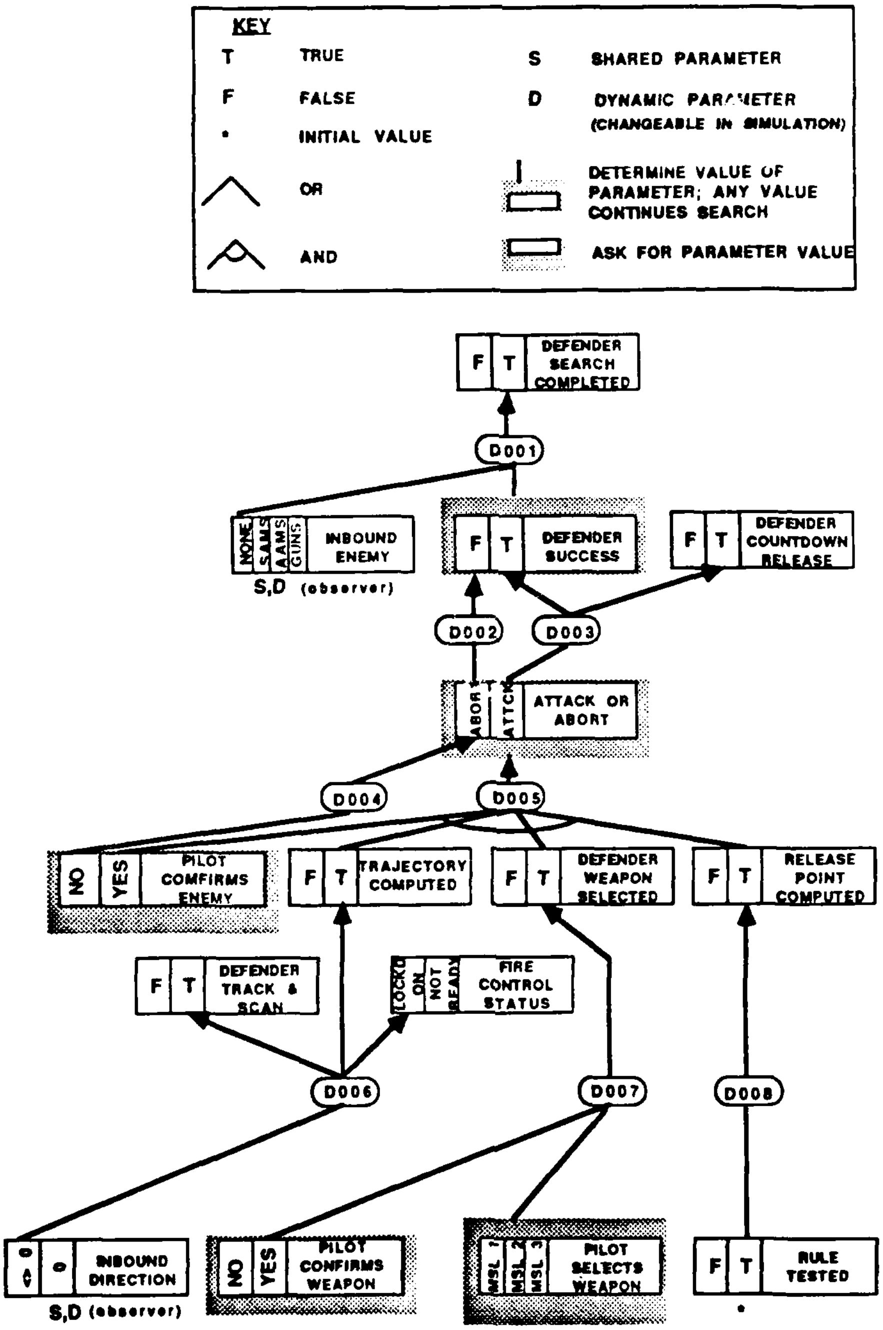
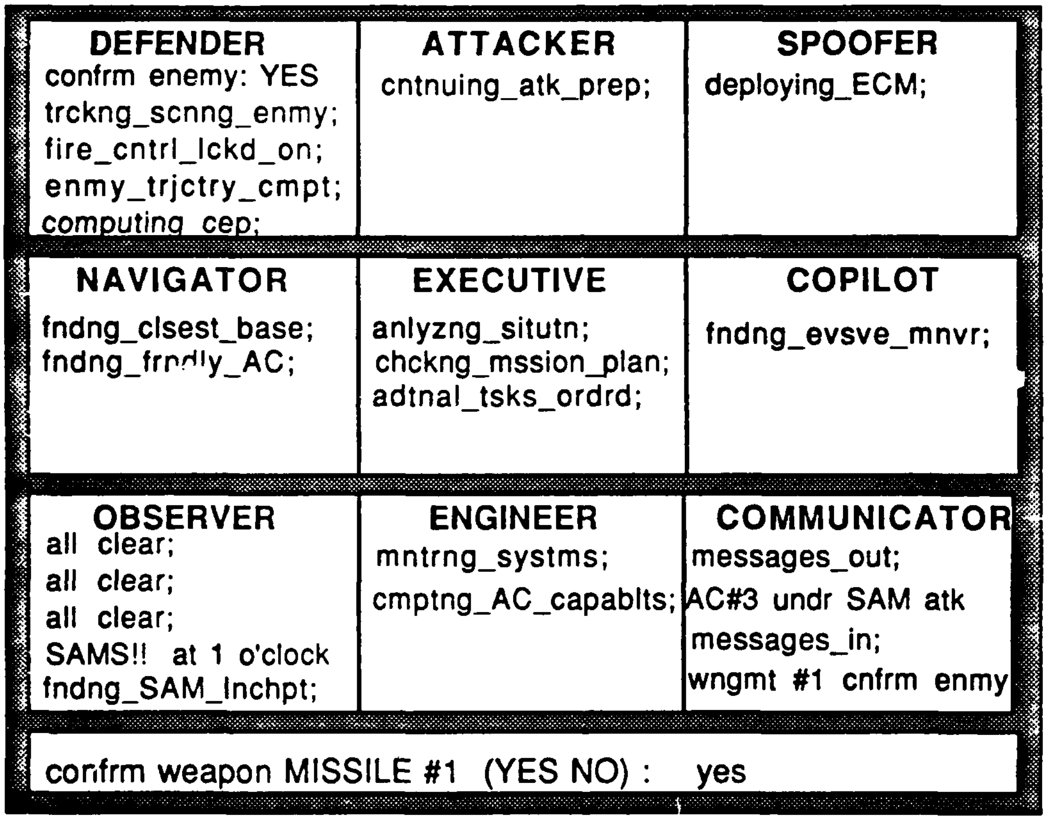

KEY
T TRUE
F FALSE
* INITIAL VALUE
OR
AND
S SHARED PARAMETER
D DYNAMIC PARAMETER (CHANGEABLE IN SIMULATION)
DETERMINE VALUE OF PARAMETER; ANY VALUE CONTINUES SEARCH
ASK FOR PARAMETER VALUE
F T DEFENDER SEARCH COMPLETED
D001
NONE SAMS AAMS GUNS INBOUND ENEMY
S,D (observer)
F T DEFENDER SUCCESS
F T DEFENDER COUNTDOWN RELEASE
D002
D003
ABORT ATTCK ATTACK OR ABORT
D004
D005
NO YES PILOT COMFIRMS ENEMY
F T TRAJECTORY COMPUTED
F T DEFENDER WEAPON SELECTED
F T RELEASE POINT COMPUTED
F T DEFENDER TRACK & SCAN
LOCKD ON NOT READY FIRE CONTROL STATUS
D006
D007
D008
INBOUND DIRECTION
S,D (observer)
NO YES PILOT CONFIRMS WEAPON
MSL 1 MSL 2 MSL 3 PILOT SELECTS WEAPON
F T RULE TESTED
*

Figure 10. Skeletal Knowledge Base of AUTOCREW Member DEFENDER.

The simulation testbed for AUTOCREW made use of a conventional personal computer. Coded keystrokes controlled the values of several knowledge-base parameters, allowing the designer to test search activity easily. The simulation included a partitioned video display, shown in Fig. 11, that presented the activity of each crew member during a simulation. The display updates on each computation cycle. Features of the display include controlled scrolling of information, flashing of crew-member windows to indicate the need for attention, and variable pen-color selection to describe the age of messages.



DEFENDER
confrm enemy: YES
trckng_scnng_enmy;
fire_cntrl_lckd_on;
enmy_trjctry_cmpt;
computing cep;
	ATTACKER
	cntnuing_atk_prep;
		SPOOFER
		deploying_ECM;

NAVIGATOR
fndng_clsest_base;
fndng_frndly_AC;
	EXECUTIVE
	anlyzng_situtn;
	chckng_mssion_plan;
	adtnal_tsks_ordrd;
		COPILOT
		fndng_evsve_mnvr;

OBSERVER
all clear;
all clear;
all clear;
SAMS!! at 1 o'clock
fndng_SAM_lnchpt;
	ENGINEER
	mntrng_systms;
	cmptng_AC_capablts;
		COMMUNICATOR
		messages_out;
		AC#3 undr SAM atk
		messages_in;
		wngmt #1 cnfrm enmy

confrm weapon MISSILE #1 (YES NO) : yes

Figure 11. AUTOCREW Display for Engineering Development.

3. LABORATORY FOR CONTROL AND AUTOMATION

The Laboratory for Control and Automation has been a focal point for this research program. As a component of Princeton's Department of Mechanical and Aerospace Engineering, the laboratory has close association with the graduate and undergraduate students of the university. At any given time, several projects normally are in progress, with specific topics chosen by participating students, faculty, and staff. These projects constitute the thesis research of graduate students (at both the M.S.E. and Ph.D. levels), the independent work projects of engineering juniors and seniors, and the research of faculty and research staff members.

The Laboratory for Control and Automation is located in the von Neumann Building, immediately adjacent to Princeton University's Engineering Quadrangle. Principal computing resources include a Symbolics 3670 LISP Machine, a Silicon Graphics IRIS 3020 Workstation, a Macintosh II computer, an IBM PS-2/80 computcr, two IBM PC-AT computers, and two IBM PC/XT computers, all of which are linked by an Ethernet communications network using the TCP/IP protocol. The most frequently used programming languages are Pascal, FORTRAN, LISP, and C; various other languages and programs, including MACSYMA and expert system shells, are available. Access to the University's IBM 3081 computer and the John von Neumann Center's Eta^{10} computer is afforded by four ports on the University's broadband network. Special-purpose computers for real-time control and data acquisition have been developed using Multibus and STD-bus architectures with various central processors. A Verbex 4000 connected-speech recognition system and several analog computers also are available for experimentation.

The Laboratory's cockpit simulator provides a single-person crew station with both conventional and advanced control devices, including the speech recognition device mentioned above. Displays for the "out-the-window" view and the control panel devices are generated entirely by computers and presented on color cathode-ray tubes. The external view is generated and displayed by the IRIS 3020 Workstation, which has 1024 x 768-pixel resolution and can perform over 85,000 3-D transformations per second. The central computing unit that performs dynamic and control calculations, accepts analog inputs, drives the panel displays, and commands the external view is a special-purpose Multibus computer employing parallel processors and controlled by an IBM PC-AT computer.

4. CONCLUSION

The principal topic of research in this effort was the development of analysis and design methods for fault-tolerant control systems, using artificial intelligence concepts to screen failure hypotheses, and more conventional control theory for reconfiguration and actual control. A Rule-Based Flight Control System for a tandem-rotor helicopter, implemented with parallel microprocessors and operating in real time, and a computer-aided controls system design tool called a Control Equation Parser were developed in the course of this research. On the basis of this research, it can be concluded that real-time applications of artificial intelligence concepts in practical control systems are easily implementable.

The sub-task entitled "Computer-Aided Heuristics for Piloted Flight" identified an expert system structure that would reduce the pilot workload associated with flying a single-seat high-performance aircraft. This research culminated in the definition of AUTOCREW, an "expert system of expert systems" based on the paradigm of a World War II bomber crew. A highly parallel structure that is amenable to the use of parallel hardware is a natural outgrowth of this work.

Both tasks made use of the Princeton Rule-Based Controller, a unique software architecture for combining procedural and symbolic processing. The knowledge base is developed in the LISP computer language and is translated with the inference engine into the Pascal language. The ease with which logic can be specified in LISP and with which Pascal code can be embedded in the knowledge-base structure makes this a highly suitable tool for task and algorithm scheduling.

REFERENCES

1. Heer, E., "Automated Decision Making and Problem Solving," NASA CP-2180, 1981.

2. Barr, A., Cohen, P., and Feigenbaum, E., *The Handbook of Artificial Intelligence*, William Kaufmann, Inc., Los Altos, 1982.

3. Nilsson, N., *Principles of Artificial Intelligence*, Tioga Publishing Co., Palo Alto, 1980.

4. Waterman, D., *A Guide to Expert Systems*, Addison-Wesley, Reading, 1986.

5. Schoenfeld, A., *Mathematical Problem Solving*, Academic Press, New York, 1985.

6. McCorduck, P., *Machines Who Think*, W.H. Freeman & Co., San Francisco, 1979.

7. Wiener, N., *Cybernetics*, MIT Press, Cambridge, 1961.

8. Brady, M., Hollerbach, J., Johnson, T., and Lozano-Perez, T., *Robot Motion: Planning and Control*, MIT Press, Cambridge, 1983.

9. Zadeh, L., "Fuzzy Sets," *Information & Control*, Vol. 8, No. 3, June 1965.

10. Ackermann, J., "Robust Control System Design," in *Fault Tolerance Design and Redundancy Management Techniques*, AGARD-LS-109, Sept 1980.

11. Harvey, C., and Pope, R., "Insensitive Control Technology Development," NASA CR-2947, Feb 1978.

12. Szalai, K., *et al*, "Flight Experience with Flight Control Redundancy Management," in AGARD-LS-109, Sept 1980.

13. Blair, J., *et al*, "Digital Flight Control Redundancy Management System Development Program," AFFDL-TR-79-3050, May 1979.

14. Willsky, A., "Failure Detection in Dynamic Systems," in AGARD-LS-109, Sept 1980.

15. Maybeck, P., "Failure Detection Through Analytic Redundancy," AFFDL-TR-78-105, Sept 1978.

16. Åstrom, K., and Wittenmark, B., "On Self Tuning Regulators," *Automatica*, Vol. 9, No. 2, Mar 1973.

17. Goodwin, G., and Sin, K., *Adaptive Filtering, Prediction, and Control*, Prentice-Hall, Englewood Cliffs, 1984.

18. Stengel, R. "Some Effects of Bias Errors in Redundant Flight Control Systems," AIAA *Journal of Aircraft*, Vol. 10, No. 3, Mar 1973.

Publications Resulting from This Research Contract

19. Stengel, R., "Artificial Intelligence Theory and Reconfigurable Control Systems," Princeton University Report No. MAE-1664, June 1980.

20. Handelman, D., "An Application of Artificial Intelligence Theory to Reconfigurable Flight Control," Princeton University Report No. MAE-1665, June 1984.

21. Huang, C., "A LISP-Based Expert System for Detecting Failures in Aircraft Systems," Princeton University Report No. MAE-1666, June 1984.

22. Loh, C., "An Application of a LISP-Based Expert System for Failure Diagnosis of the CH-47 Flight Control Hydraulic System," Princeton University Report No. MAE-1741, Mar 1986.

23. Loh, C., "A Prototype Maintenance Expert System for the CH-47 Flight Control Hydraulic System," Princeton University Report No. MAE-1751, Apr 1986.

24. Stengel, R., "Artificial Intelligence and Reconfigurable Control Systems," *SIGART Newsletter*, No. 92, Apr 1985.

25. Handelman, D., and Stengel, R., "Combining Quantitative and Qualitative Reasoning in Aircraft Failure Diagnosis," in *Proceedings of the 1985 AIAA Guidance, Navigation, and Control Conference*, AIAA Paper No.

85-1905CP, Aug 1985.

26. Huang, C., and Stengel, R., "A Symbolic Parser for Control Equations," in *Proceedings of the Second Workshop on Computer-Aided Control System Design*, July 1986. (to appear in the *Transactions of The Institute of Measurement and Control.*)

27. Handelman, D., and Stengel, R., "A Theory for Fault-Tolerant Control Combining Expert System and Analytical Redundancy Concepts," in *Proceedings of the 1986 AIAA Guidance, Navigation, and Control Conference*, AIAA Paper No. 86-2092CP, Aug 1986. (revised version to appear in the AIAA *Journal of Guidance, Control, and Dynamics* entitled "Combining Expert System and Analytical Redundancy Concepts for Fault-Tolerant Control").

28. Huang, C., and Stengel, R., "Failure Model Determination in a Knowledge-Based Control System," *Proceedings of the 1987 American Control Conference*, Minneapolis, June 1987.

29. Handelman, D., and Stengel, R., "An Architecture for Real-Time Rule-Based Control," *Proceedings of the 1987 American Control Conference*, Minneapolis, June 1987.

30. Huang, C., and Stengel, R., "Restructurable Control Using Proportional-Integral Implicit Model Following," *Proceedings of 'he1987 AIAA Guidance, Navigation, and Control Conference*, Monterey, Aug 1987.

31. Handelman, D., and Stengel, R., "Rule-Based Mechanisms for Intelligent Adaptive Control," *Proceedings of the 1988 American Control Conference*, Atlanta, June 1988.

32. Handelman, D., and Stengel, R., "Perspectives on the Use of Rule-Based Control," to be presented at the IFAC Workshop on Artificial Intelligence in Real-Time Control, Swansea, England, Sept 1988.

33. Suntharalingam, P., "Application of Linear-Quadratic Model-Following Controllers in Restructurable Control Systems," Princeton University M.S.E. thesis, Aug 1988 (est).

34. Handelman, D., "A Theory for Rule-Based Control," Princeton University Ph.D. thesis, Aug 1988 (est).

35. Huang, C., "Intelligent Fault-Tolerant Control Systems," Princeton University Ph.D. thesis, Aug 1988 (est).

36. Suntharalingam, P., "A Test on the Reliability and Performance of the Verbex Series 4000 Voice Recognizer, Princeton University MAE Report No. MAE-1748, Sept 1985.

37. Belkin, B., "A Demonstration Expert System for Implementing Emergency Procedures in a High-Performance Fighter Aircraft," Princeton University MAE Report No. MAE-1749, Apr 1986.

38. McCaffrey, J., "F-18 Hornet Cockpit Display Simulation," Princeton University Senior Independent Work Project Progress Report, Jan 1987.

39. McCaffrey, J., "F-18 Hornet Horizontal Situation Indicator Display Simulation," Princeton University Senior Independent Work Project Report, May 1987.

40. Belkin, B., "AUTOCREW Implementation: Inbound Surface-to-Air Missile Simulation," Princeton University MAE Report (no number), May 1988.

41. Belkin, B., and Stengel, R., "Cooperative Rule-Based Systems for Aircraft Control," *Proceedings of the 26th IEEE Conference on Decision and Control*, Los Angeles, Dec 1987.

42. Belkin, B., "Cooperative Rule-Based Systems for Aircraft Navigation and Control," Princeton University M.S.E. thesis, Aug 1988 (est).

END
DATE
FILMED
DTIC
9-88